Name: Compute Cloud

Contact: Michael Behrens, R2AD

Description: A cloud provider implements a RESTful API for provisioning, executing, and monitoring of tasks.

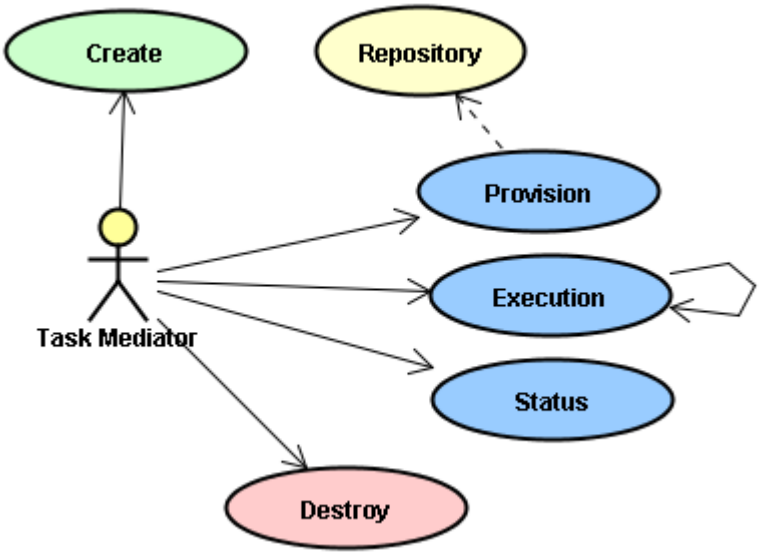
Functional Requirements

> **Secure**: API must be secured to ensure that only authorized identities are permitted to use the API.
>
> **Resource**: An endpoint must be created for external monitoring, status, and auditing of the task. This endpoint would be responsive to RESTful calls supporting AJAX and other clients.
>
> **Scripted**: The target system needs to understand and process directives which would be provided with the task. These directives would include the ability to pull binaries or data onto the system, run executables, and status the system resources.

Non-functional Requirements

> **Single Compute Method**: The resultant service should be the same service that can be used for many other purposes. It could be used for monitoring of system health, system life-cycle management, system patching, and configuration changes. If this was the only service on the system initially, it could then be used to build up the other services in a plug-in manner.

Compute Cloud Use Case - Brainstorming Ideas

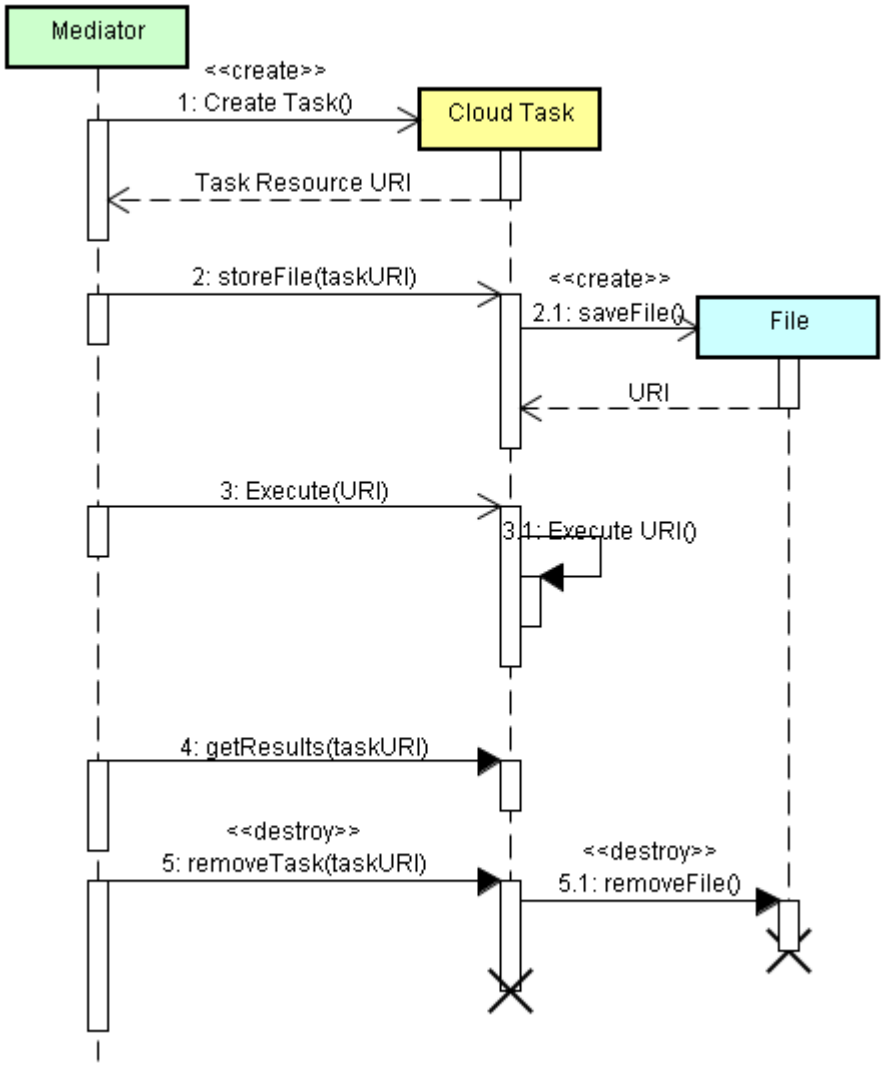


Given an identified group of machines, data and/or executable is provisioned to each machine and then processed or accessed. For example, a patch update could be sent to all

machines. At a specified time or time offset, the patch script and/or binaries are executed. The status and the results of the execution can be collected if needed and made available to support reports and alerts.

In order to know which machines, a query task (or alternative mechanism) could be used to determine the machines identity and IP address. For instance, in the case of applying a patch or new version of software, the query might be one which asks which machines have version x.y.z of some software package installed. A more complicated scenario could add a repository that might also provides inventory information. This query itself could be a compute cloud execution event, meaning that one might want to send a compute task to all machines which executes a script on all machines to determine if a set of package is installed. Then the return results could then be used by the follow-on task activity (cloud task) to provision, execute, and monitor the patch update.

If a repository were used, then tasks could be set to it to update its contents, thereby providing rudimentary replication/synchronization between them as might be needed based on authorization and need.



A task mediator would be needed interact with the user (client) and to track the status of the entire workflow. This does not need to be part of the cloud system itself, as it could just be a mashup running within the context of a browser.